[10191/4188]

DEVICE AND METHOD FOR FORMING A SIGNATURE

Background Information

MISR (Multiple Input Signature Register) circuits, such as described in the publication "Built-In Test for VLSI: Pseudorandom Techniques" by Paul H. Bardell, William H. McAnney, and Jacob Savir, pp. 124 et seq., are used for forming

5   signatures. In circuits of this type, a given number of shift registers is provided to which data to be tested is applied in a sequence. The data occurring simultaneously is injected and shifted forward by the shift registers in a predefined cycle. After a precisely defined number of data words and cycles, there is a signature value in the shift registers which is testable and comparable with a previously known signature

10   value. To test a sequence, and thus the data applied, for errors, it is sufficient to compare the signature value obtained with the expected signature value. The previously known signature values may also be determined in this way.

Problems occur with the method and the device of the related art when at a point in

15   time T there is an error at a certain input, because an incorrect value is then written initially into the shift register affected. The computed end signature will therefore differ from the expected signature. However, if an additional error occurs at a subsequent point in time T+1 at a subsequent, in particular directly subsequent, input, the original error at the first input is compensated again, after being shifted by

20   the shift registers, using a number of cycles which corresponds to the distance between the inputs and points in time, using one cycle in particular, if there is no feedback junction of the MISR, i.e., the signature generator circuit, in between. Errors occurring at such problem points in time and data word positions are thus not detected in forming the signature.

25

One option for taking precautions during the input to avoid this problem is to supply the inverse data word following a data word, so that an error is not compensated but detected in each case. However, this procedure doubles the number of required operations and cycles.

30

SUBSTITUTE SPECIFICATION

It is thus evident that the related art was unable to provide results that are optimum from every point of view, making the object necessary to develop an improved device and an improved method for mastering the above-named problems when forming signatures.

5

Summary Of The Invention

The present invention is directed to a device and a method for forming a signature, a predefined number of shift registers (shift register with a predefined number of bit positions, represented by flip-flops) being provided, to which input data to be tested

10 is applied bit-by-bit and in parallel as successive data words and which serially shift the input data forward in a predefinable cycle, and a signature is formed in the shift registers after a certain number of data words and cycles, a code generator which generates at least one additional bit position in at least one additional shift register from each data word in the signature also being advantageously provided. This

15 means that the MISR is advantageously extended by at least one bit position; this bit position is obtained from the complete data word applied at the time and is also included in the signature. This makes it advantageously possible to master the above-mentioned problem without performing a plurality of additional operations and cycles when forming the signature.

20

Error masking in the event of the above-mentioned multiple errors is thus prevented using minimum additional circuit complexity.

It is furthermore advantageous that the individual shift registers are connected by

25 antivalence points, i.e., XOR gates, and the individual bit positions are also inserted via these antivalence points.

It is also conceivable to use an equivalence point, i.e., a negated XOR instead of an antivalence gate, i.e., an antivalence point to insert the individual bit positions of the

30 data words and the at least one bit position of the code generator into the corresponding shift registers.

SUBSTITUTE SPECIFICATION

The code generator is advantageously designed in such a way that it implements an ECC (Error Check and Correction) code such as, for example, a Hamming code, a Berger code, or a Bose-Lin code, etc., to input the number of bit positions corresponding to the particular ECC code into a corresponding number of additional

5 shift registers to form the signature. In the most generic case, a code generator table (hard-wired or in software form) may be used to assign a desired code pattern of any desired length to a given input pattern of data words or bits. In the simplest case, the code generator is advantageously designed in such a way that it forms a parity bit and inputs it into an additional (bit position of the) shift register (represented by a

10 flip-flop or another memory element).

Brief Description Of The Drawings

Figures 1-4 illustrate the device and method according to the present invention .

15 Detailed Description

Figure 1 shows an MISR circuit having shift registers 100 through 105 and antivalence, i.e., XOR, gate points 106 through 111. A modular type corresponding to the feedback is illustrated here. Input 0, input 1, input 2, input 3, input 4, and input n-1, which correspond to the respective bit positions of the data words applied are

20 inserted into the shift registers and input and shifted through in a predefined cycle. In the shift registers this then results in states X0, X1, X2, X3, X4, and Xn-1, where n is a natural number greater than zero, in this concrete example at least 6.

Figure 2 also shows an MISR circuit having shift registers 100 through 105 and

25 antivalence, i.e., XOR, gates 106 through 111. Furthermore, two additional XOR gates 111 and 113 are shown which in this example are located downstream from shift register 100 and shift register 102. This is therefore a standard type of an MISR; the insertion points, i.e., antivalence gates 112 and 113, as well as their number, may be freely selected in the MISR. Also in this case, inputs 0 through n-1

30 are shown, as well as the states of shift registers X0 through Xn-1 where n ∈ N.

Figure 3 shows three data words DW1, DW2, and DW3, which are to be applied to inputs 0 through n-1 in this order. The individual bit positions are labeled BS0

through BSn-1. If there is an error F in data word DW1, for example, intended for input 1 at point in time T and also in data word DW2, i.e., at input 2 at a later point in time T+1, these errors are compensated after shifting by one cycle in the MISR. The same applies to other error configurations resulting in compensation due to their

5   time of insertion or position in the data word and the corresponding input.

In Figure 4, the MISR is now extended by an i bit code generator 407. Here i, also as a natural number greater than zero, stands for the number of bits inserted into the MISR by the code generator according to the code used or the ECC code in the

10   code generator. A number of shift registers (flip-flops), labeled 408 here, corresponding to this number i of the bit positions output by the code generator, is also provided in addition to the MISR . In the simplest case, a parity bit is formed here, so that only one additional shift register and one additional input -1 is then provided.

15

The position in the MISR where the at least one additional shift register or the at least one additional insertion point, i.e., antivalence or equivalence point, is inserted is freely selectable and is shown here as an example only. This means that here also in Figure 4 the usual shift registers 100 through 105 are shown, at least one

20   additional shift register 408 being provided. The inputs of the device according to the present invention, input 04, input 14, input 24, input 34, input 44, and input (n-1)4, are connected here not only to the antivalence points, i.e., the XOR gates, but also to the i bit code generator. Thus, in the predefined cycle, additional information, dependent on the code used (ECC in particular), is generated from each incoming

25   data word and input into an appropriate number of shift registers. In this example, elements 400 through 406 are provided as XOR gates, in our example the usual states X0, X1, X2, X3, and Xn-1 of the shift registers resulting in addition to input -i and state X-i of shift register 408. The additional arrows as the output of i bit code generator 407 indicate that in another embodiment even more than just one

30   additional bit position are written into the MISR, depending on the code used.

Thus, for example, if a Hamming code is used with ECC for single-error correction and 4 bits of useful data, a 3-bit correction code results. For ECC single-error

correction and 8 bits of useful data, 4-bit correction codes result. For 16 bits of useful data, 5-bit correction codes, and for 32 bits of useful data 6-bit correction codes result. This means that, in general, $2^k>=m+k+1$, where m is equivalent to the number of useful bits as a natural number greater than zero, and k is equivalent to

5 the number of code bits or correction bits, i.e., the correction code also as a natural number. If double-error detection is also to be performed, one additional bit is to be provided for the correction code.

For example, if a Berger code is used, 3 code bits and 5 states are to be provided

10 for 4 bits of useful data; for 8 bits of useful data and 9 states, 4 additional code bits are to be provided. For 16 bits of useful data, 5 additional code bits for 17 states, and for 32 bits of useful data 6 additional code bits for 33 states are to be provided. Here, in general, $2^k>=m+1$ or $k>=ld(m+1)$, where m is the number of useful bits of the data and k is the number of code bits, i.e., the correction code.

15

Further codes such as the Bose-Lin code may also be used, in which case the number of coding bits is the same as in the Berger code, but the check bits used are either only modulo 4 or modulo 8.

20 The number of code generator outputs, i.e., additional inputs –i where i = 1 through k ∈ N, and the number of shift registers and gates provided also corresponds to the number of these coding bits k.

The MISR is thus extended by at least one position by obtaining at least one parity

25 or other code from the original data input 0 through input n-1, and is also included in the signature, shown in this example of Figure 4 for the modular type (Figure 1). Of course, the same applies to the standard type (Figure 2). The code generator may thus be a parity generator where i = 1, in this case one additional flip-flop being required. In the event where an error occurs, for example, at input 3, a modified

30 value is additionally supplied at input -1, i.e., the parity input. To mask this value in the event of an error, an error is required both at input 4 and at input 0 specifically in the next cycle. This means that there is a higher Hamming distance here and the

likelihood of masking is considerably reduced due to the required precise timing in the event of error masking via double error.

The Hamming distance may be further increased as desired using more code bits as mentioned above. If, instead of antivalence, an equivalence gate is used for insertion, still a considerably lower error masking probability, although a slightly lower redundancy, is achieved than in the related art.

As another option, code generator 407 may also use assignment by table, i.e., a code generator table causes a predefined number of code bits to be inserted into a corresponding number of shift registers as a function of the incoming bit combination of the data word. Any desired assignment of incoming data bits to output coding bits is possible by using such a code generator table.

In the serial alternative, a switching means S which interrupts the feedback line and allows the registers to be serially read is provided for reading the signature formed from the MISR. Another option is, as indicated by the letter P and the dashed line, to output the shift registers in parallel and thus to output the signature from the MISR all at once to compare it with a corresponding expected signature.

This means that the present invention represents a considerably higher safety factor than a conventional MISR, while using less resources than a constantly required inversion of the data words for compensating an error masking.

Therefore, the present invention is usable in all applications critical with regard to safety, in particular in the automobile industry such as in brake controls (ABS, ASR, ESP, etc.), steer-by-wire, brake-by-wire, i.e., in general x-by-wire, airbag, engine control, transmission control, etc. The present invention may also be used in microcontrollers or other semiconductor structures in testing, as well as in all BIST (built-in self-test) structures and for optimizing production testing.